RECEIVED
CENTRAL FAX CENTER
OCT 27 2008

Amendments to the Claims:

Please amend Claims 1, 2, 4-8, and 10; cancel Claim 9; and add new Claims 11-21, as shown below.

1. (Currently Amended): A method for implementing a two-phase commit protocol, comprising:

dispatching a first ~~two-phase commit protocol~~ prepare operation from a first server thread to a second server thread, wherein the first ~~two-phase commit protocol~~ prepare operation is associated with a first resource and a ~~first~~ prepare phase ~~of two-phase commit protocol~~;

processing a second ~~two-phase commit protocol~~ prepare operation by the first server thread in parallel to the first prepare operation being processed by the second server thread, wherein the second ~~two-phase commit protocol~~ prepare operation is associated with a second resource and the ~~first~~ prepare phase ~~of two-phase commit protocol~~; ~~and~~

determining that the prepare phase ~~the first two-phase commit protocol operation~~ is complete;

dispatching a first commit operation from the first server thread to a third server thread, wherein the first commit operation is associated with the first resource and a commit phase;

processing a second commit operation by the first server thread in parallel to the first commit operation being processed by the third server thread, wherein the second commit operation is associated with the second resource and the commit phase; and

after determining that the commit phase is complete, writing results of the commit phase to a transaction log.

2. (Currently Amended): The method of claim 1 further comprising:

selecting an idle server thread to process the first prepare ~~two-phase commit protocol~~ operation.

3. (Original): The method of claim 2, wherein selecting includes:

determining available server threads in a server.

4. (Currently Amended): The method of claim 3 wherein a thread pool manager determines the available server threads in the server.

5. (Currently Amended): The method of claim 1 further comprising:

reporting results of the prepare phase ~~first and second two-phase commit protocol~~ to a log.

6. (Currently Amended): A method for processing N two-phase commit protocol operations, comprising:

processing N ~~two-phase commit protocol~~ prepare operations in a first server thread, wherein each of the N prepare operations are associated with a prepare phase, wherein the processing for each prepare operation of N-1 of the prepare operations ~~two phase commit protocol operation~~ includes:

dispatching the prepare ~~two-phase commit protocol~~ operation to a second server thread if a second server thread is determined to be available; and

processing the prepare ~~two-phase commit protocol~~ operation in the first server ~~primary~~ thread if no second server thread is determined to be available; ~~and~~

processing ~~the~~ a remaining prepare ~~two phase commit protocol~~ operation in the first server thread;

determining that the prepare phase is complete;

processing N commit operations in a first server thread, wherein each of the N commit operations are associated with a commit phase, wherein the processing for each commit operation of N-1 of the commit operations includes:

dispatching the commit operation to a second server thread if a second server thread is determined to be available;

processing the commit operation in the first server thread if no second server thread is determined to be available;

processing a remaining commit operation in the first server thread; and

after determining that the commit phase is complete, results of the commit phase are written to a transaction log.

7. (Currently Amended): The method of claim 6 wherein dispatching ~~the~~ each two-phase commit protocol operation to a second server thread includes:

determining available server threads in a server; and

selecting one of the available server threads to be the second server thread.

8. (Currently Amended): The method of claim 7 wherein a thread pool manager determines the available server threads in the server.

9. (Cancelled)

10. (Currently Amended): The method of claim 6 further comprising:

reporting results of the ~~first and second~~ N ~~two-phase commit protocol~~ prepare operations associated with the prepare phase to a log.

11. (New): The method of claim 1, wherein a dedicated thread pool is used for parallel transaction operations.

12. (New): The method of claim 1, wherein a transaction manager implements Java Transaction API.

13. (New): The method of claim 1, wherein the first resource is an XA resource.

14. (New): The method of claim 6, wherein each prepare operation of N-1 of the prepare operations is associated with an XA resource.

15. (New): The method of claim 6, wherein a dedicated thread pool is used for parallel transaction operations.

16. (New): The method of claim 6, wherein a transaction manager implements Java Transaction API.

17. (New): A system, comprising:

a dedicated thread pool for parallel transaction operations, including:

a first server thread, wherein the first server thread processes a first prepare operation by the first server thread, and wherein the first prepare operation is associated with a first resource and a prepare phase;

a second server thread, wherein the first server thread dispatches a second prepare operation to the second server thread, wherein the second prepare operation is associated with a second resource and the prepare phase and processed by the second server thread in parallel to the first prepare operation being processed by the first server thread;

a transaction manager that implements Java Transaction API, wherein after the transaction manager determines that the prepare phase is complete, the first server thread processes a first commit operation associated with the first resource, and the first server thread dispatches to a third server thread a second commit operation associated with the second resource, wherein the second commit operation is processed by the third server thread in parallel to the first commit operation being processed by the first server thread; and

a transaction log, wherein after the commit phase is complete, results of the commit phase are written

to the transaction log.

18. (New): The system of claim 17 further comprising:
a thread pool manager, wherein the thread pool manager determines available server threads.

19. (New): The system of claim 17 wherein the transaction log records results of the prepare phase.

20. (New): The method of claim 1, wherein all of the prepare operations and all of the commit operations are part of a single transaction.

21. (New): The method of claim 6, wherein all of the prepare operations and all of the commit operations are part of a single transaction.